Heterogeneous Federated Learning on Arbitrary Graphs

Huiyuan Wang[*], Xuyang Zhao[*], and Wei Lin[*]

Abstract

Federated learning has emerged as a significant focus in distributed machine learning practice, where algorithms are trained across multiple decentralized devices without sharing local data. In this work, we consider parameter estimation in federated learning with heterogeneity in communication and data distribution, and with limited computational capacity of devices. We model the distribution heterogeneity using a latent graph, in which devices are adjacent if, and only if, they share the same target parameter. With knowledge of a surrogate graph, we propose to jointly estimate parameters for all devices under the M-estimation framework with the fused Lasso regularization. We provide a general statistical guarantee for our regularized estimator under arbitrary surrogate graphs, which can be further calibrated to convergence rates for various specific setups. If the surrogate graph satisfies a graph fidelity condition, then our estimator is optimal as if we could aggregate all samples sharing the same distribution. Otherwise, we propose an edge selection procedure via multiple testing to ensure the optimality. To reduce the burden of local computation, a decentralized stochastic version of ADMM, termed FedADMM, is provided with convergence rate $O(T^{-1}\log T)$, where T denotes the number of iterations. Our algorithm transmits only parameters along edges of $\mathcal{G}$ at each iteration without requiring a central machine. FedADMM is further extended to the case where devices are randomly inaccessible during the training process with a similar convergence guarantee. The computational and statistical efficiency of our method is evidenced by simulation experiments and the 2020 US presidential election data set.

Keywords: Federated learning, network lasso

[*] School of Mathematical Sciences and Center for Statistical Science, Peking University, Beijing, China

Address for correspondence: Wei Lin, School of Mathematical Sciences and Center for Statistical Science, Peking University, Beijing 100871, China (E-mail: weilin@math.pku.edu.cn).

1 Introduction

In recent years, the proliferation of intelligent devices such as smartphones and wearable technology has facilitated unprecedented access to personal data. This wealth of information has been actively utilized to develop personalized models, with applications ranging from predicting text inputs and scheduling traffic patterns, to recognizing emotions and monitoring health (Cui et al. 2021; Accettura et al. 2013; Strain et al. 2020; Wu et al. 2020). However, the analysis of personal data presents unique challenges. First, personal data are naturally stored in a distributed manner, and the total number of devices greatly exceeds the per-device sample size (Mach and Becvar 2017), making it impossible to aggregate and analyze all the data on a single machine. Legislative regulations add another layer of complexity by prohibiting data sharing across devices (Voigt and von dem Bussche 2017). In addition, the computational resources of intelligent devices are typically limited, restricting processing capabilities to mini-batches of samples at a time. Lastly, as no two users are identical (Li and Meng 2021), data stored on different devices essentially stem from distinct distributions. These critical issues have propelled the rise of federated learning in practical applications (Konečnỳ et al. 2016).

Federated averaging (FedAvg), arguably the most prevalent algorithm in federated learning, addresses some of these challenges (McMahan et al. 2017). A typical FedAvg iteration includes: (1) running several steps of stochastic gradient descent (SGD) in parallel on an independently sampled subset of devices, (2) sending the updated local parameters to a central server, and (3) averaging these parameters using prespecified weights. Notably, FedAvg aligns with data privacy regulations, requiring only parameter transmission. It effectively manages large numbers of devices through device sampling and adapts to the limited computational capabilities of devices by restricting the number of SGD steps performed locally. Importantly, the output of FedAvg converges to the pooled estimator as if all data were aggregated on a single machine (Li et al. 2020), a desirable feature when our goal is to infer the whole population.

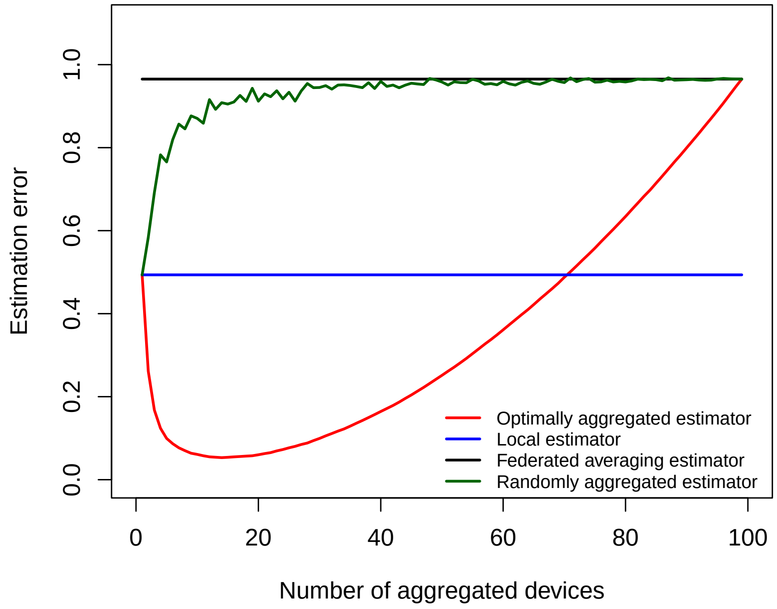

Figure 1: Estimation error as the number of aggregated devices increases, based on 10^3 independent repeated experiments.

While the FedAvg offers a template for overcoming algorithmic barriers in analyzing personal data, it unfortunately overlooks the unique characteristics of individuals. This oversight can potentially lead to erroneous results, as demonstrated below. Suppose that each device, enumerated by $k = 1, \ldots, 100$, contains two independent observations from the distribution $N(k/50, 1)$. Our goal is to estimate the population mean of the first device. Here, as true parameters differ across local devices, aggregating data from other devices can introduce bias. Notably, the bias can offset the benefits of variance reduction gained through aggregation. This aggregation–heterogeneity trade-off further explains why the FedAvg estimator performs poorly (Zhao, Wang, and Lin 2023). In fact, from the viewpoint of transitional inference (Hankinson 1987), the optimal strategy is to aggregate data from the devices whose target parameters are as close to that of the first device as possible, giving rise to the "optimally aggregated estimator." As the number of aggregated devices increases, we observe that the squared error for the optimally aggregated estimator has a U shape, which confirms the trade-off between aggregation and heterogeneity.

Interestingly, for optimal estimation it is necessary to have knowledge of device rankings based on the distance between their true parameters. In our example, the performance of the “Randomly aggregated estimator,” which is derived by randomly aggregating devices, degrades rapidly as the number of aggregated devices increases. In fact, without the prior information of device rankings, Zhao, Wang, and Lin (2023) showed that no estimator can achieve a convergence rate faster than the local estimator under certain smoothness assumptions of the distribution heterogeneity. This intrinsic inability to adapt underscores the importance of acquiring the precise ranking information. However, this acquisition is a considerable challenge, particularly when the total number of devices is large since each local device has its unique ranking of other devices.

To relax the requirement of knowing accurate device rankings, in this study we propose the use of adjacency structures to model distribution heterogeneity. Specifically, we posit a latent graph $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E}_0)$, where the node set $\mathcal{V}$ represents all the devices and the edge set $\mathcal{E}_0$ comprises all pairs of devices that share the same target parameter. While these adjacency structures impose stronger restrictions on target parameters, we do *not* require precise knowledge of $\mathcal{G}_0$, but rather a surrogate graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with potential discrepancies. Interestingly, $\mathcal{G}$ must encapsulate some relevant knowledge about $\mathcal{G}_0$. Otherwise, we prove in Proposition 1 that it is impossible to adapt to unknown adjacency structures.

To leverage the prior knowledge encapsulated by $\mathcal{G}$, we incorporate a network-fusion regularization to be defined in Section 3 into a general M-estimation framework, which encourages an equal estimate for the target parameters of devices connected in $\mathcal{G}$. We further give a deterministic risk bound in Theorem 2 for the network-fusion penalized M-estimator which holds irrespective of the distribution of score functions. For sub-Gaussian score functions, the deterministic risk bound has an explicit form $O_P\big[p\{K(\mathcal{G}) + S\}(\log V)/(nV)\big]$ in Theorem 3, where p is the dimension of parameter space, $K(\mathcal{G})$ denotes the number of connected components of $\mathcal{G}$, S is the number of edges present in $\mathcal{G}$ but absent from $\mathcal{G}_0$, n denotes per-device sample size, and V denotes the total device counts. Our theoretical bound also exhibit the aggregation–heterogeneity trade-off: to achieve a smaller S, edges in $\mathcal{G}$ need to be eliminated, resulting in a larger value of $K(\mathcal{G})$. Moreover, if the quantity $K(\mathcal{G}_0)/\{K(\mathcal{G})+S\}$, which we term graph fidelity, does not tend to zero as the graph size grows, then our proposed estimator attains the optimal rate, as if we could aggregate all identically distributed samples separately. We then propose an edge selection procedure through multiple testings so that the graph fidelity is maximized. We prove the model selection consistency in Theorem 5.

After addressing distribution heterogeneity, we propose FedADMM to overcome algorithmic challenges. Specifically, FedADMM is a decentralized, stochastic version of the Alternating Direction Method of Multipliers (ADMM) algorithm. Each FedADMM iteration comprises a node optimization step and an edge communication step, both of which do not require the coordination of a central server. At the node optimization step, local devices need only perform one-step SGD, which is feasible for devices with limited computational capacity. At the edge communication step, only parameters are transmitted among connected devices. This device-to-device algorithm is effective for handling a large number of devices, especially when the communication graph is sparse (Boyd et al. 2006). We prove in Theorem 6 that FedADMM converges with the rate $O(T^{-1}\log T)$, where T denotes the total number of iterations. We further demonstrate in Corollary 7 that our algorithm attains the same convergence rate under malicious random block of devices in the optimization process.

1.1 Related Work

Federated learning is indeed a subset of distributed learning, albeit with additional constraints derived from practical scenarios. Existing distributed statistical methods often work under the assumption of independently and identically distributed samples and cover a wide array of topics such as M-estimation, nonparametric regression, principal component analysis, and Bayesian methods (Zhang, Duchi, and Wainwright 2013; Battey et al. 2018; Fan et al. 2019; Banerjee, Durot, and Sen 2019; Fan, Guo, and Wang 2021; Jordan, Lee, and Yang 2019). Some studies explore heterogeneous distributed learning, assuming that the impact of covariates on outcomes can be split into a common effect and device-specific effects (e.g., Zhao, Cheng, and Liu 2016; Duan, Ning, and Chen 2021). However, this requires prior knowledge of which covariates give rise to device-specific effects.

Additionally, Richards, Negahban, and Rebeschini (2021) investigated scenarios where each node holds a high-dimensional linear model, and two nodes are linked if the difference in their target parameters is also sparse. Nonetheless, the minimal per-device sample size required therein grows linearly with the graph size, even in the noiseless setting. More recently, Zhang, Liu, and Zhu (2022) modeled heterogeneity in distributed linear regressions via a latent cluster structure but assumed this latent structure could be consistently estimated. Our approach differs as we provide non-asymptotic analyses based on a surrogate graph, offering broader applicability. Moreover, while these distributed statistical methods are valid under respective assumptions and require few communication rounds, they necessitate precise task solving on each local device, making them more suitable for multi-center research (Sidransky et al. 2009).

Another related area to this study is the network Lasso penalized estimation, or trend filtering on graphs (Hallac, Leskovec, and Boyd 2015; Hütter and Rigollet 2016; Wang et al. 2016, among others). These methods assume parameter sparsity over a predefined graph or network. For Gaussian mean estimation with total variation regularization, Hütter and Rigollet (2016) derived a sharp convergence rate. Hallac, Leskovec, and Boyd (2015) leveraged distributed ADMM to solve optimization problems analogous to ours. Our work generalizes the former to general M-estimation and the latter to a stochastic federated settings.

1.2 Organization of This Paper

The rest of the paper is organized as follows. We present some necessary notation and problem setup in Section 2. Section 3 contains details of the network-fusion penalized estimator, its theoretical properties, and the edge selection procedure through multiple testings. In Section 4, we introduce the FedADMM together with its extension to derive our estimator and show their algorithmic consistency. Section 5 consists of simulations, and a real-world data analysis is included in Section 6.

2 Preliminaries

We first introduce some notation used in this article.

2.1 Notation

The ℓ_q-norm on $\mathbb{R}^p$ norm is denoted by $\|\cdot\|_q$ for $q \geq 1$. Define $\mathbb{B}(\mathbf{a}; r) = \{\mathbf{x} : \|\mathbf{x}-\mathbf{a}\|_2 \leq r\}$ as the ball in $\mathbb{R}^p$ with center $\mathbf{a}$ and radius r. Let $\mathcal{S}$ be any set, and $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$. For a matrix $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_q)^T \in \mathbb{R}^{q\times p}$, we define the ℓ_1/ϕ-norm of $\mathbf{A}$ as $R(\mathbf{A}) = \sum_{j=1}^{q} \phi(\mathbf{a}_j)$, where $\phi\colon \mathbb{R}^p \to [0, \infty)$ is a norm on $\mathbb{R}^p$. The notation $(\mathbf{a}_k : k \in \mathcal{S})^T$ represents the submatrix formed by the rows of $\mathbf{A}$ indexed by $\mathcal{S}$. For a symmetric matrix $\mathbf{A}$, $\lambda_{\max}(\mathbf{A})$ represents its maximum eigenvalue, and $\lambda_{\min}(\mathbf{A})$ represents its minimum eigenvalue. If $\mathbf{A}$ is positive semi-definite, $\lambda^{+}_{\min}(\mathbf{A})$ denotes its smallest nonzero eigenvalue.

The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by its node set $\mathcal{V} = \{1, \ldots, V\}$ and edge set $\mathcal{E}$. The cardinalities of $\mathcal{V}$ and $\mathcal{E}$ are denoted by V and E, respectively. For an edge $e = (i, j) \in \mathcal{E}$, let e^+ represent $\max(i, j)$ and e^- represent $\min(i, j)$. The signed incidence matrix with respect to $\mathcal{E}$ is denoted by $\mathbf{D} \in \{-1, 0, 1\}^{E\times V}$, where the (e, i)th entry is given by $D_{ei} = I(i = e^+) - I(i = e^-)$ for all $e \in \mathcal{E}$ and $i \in \mathcal{V}$, with $I(\cdot)$ being the indicator function. The Moore-Penrose inverse of $\mathbf{D}$ is denoted by $\mathbf{D}^{\dagger}$. When referring to connected components of a graph, we use the notation $\mathcal{C}_1, \ldots, \mathcal{C}_K$, where K represents the number of connected components in $\mathcal{G}$. To emphasize the dependence of K on $\mathcal{G}$, we sometimes write $K(\mathcal{G})$.

2.2 Problem Setup

Suppose that on the device u we can observe n independent copies $\mathbf{z}_k^{(u)}$ from an unknown distribution P_u for all $u \in \mathcal{V}$. Under a general M-estimation framework, the target parameter $\boldsymbol{\theta}_u^*$ for the device u is defined by

$$\boldsymbol{\theta}_u^* = \underset{\boldsymbol{\theta}\in\boldsymbol{\Xi}_u}{\operatorname{argmin}}\, M_u(\boldsymbol{\theta}) \equiv \mathbb{E}_{P_u}\{m_u(\mathbf{z};\boldsymbol{\theta})\}. \tag{1}$$

We refer to $\boldsymbol{\Xi}_u$ as the natural parameter set, which is a bounded subset of $\mathbb{R}^p$ containing interior points. Throughout this article, we take the dimension p of target parameters as fixed. This framework includes a wide spectrum of statistical models as a special case, including mean estimation, linear regression models, logistic regression models, Gaussian graphical models, and additive hazard models. See Examples 1–5 in the Supplementary Material for details. The Hessian matrix of $M_u(\boldsymbol{\theta})$ is denoted by $\mathbf{H}_u(\boldsymbol{\theta})$. To ensure the identifiability of target parameters, we need certain regularity conditions.

Condition 1 (Identifiability of target parameters)**.** The population Hessian matrices have bounded eigenvalues,

$$\underline{\lambda} \le \min_{u\in\mathcal{V}} \inf_{\boldsymbol{\theta}\in\boldsymbol{\Xi}_u} \lambda_{\min}\{\mathbf{H}_u(\boldsymbol{\theta})\} \le \max_{u\in\mathcal{V}} \sup_{\boldsymbol{\theta}\in\boldsymbol{\Xi}_u} \lambda_{\max}\{\mathbf{H}_u(\boldsymbol{\theta})\} \le \overline{\lambda}.$$

Condition 1 is imposed on the population level, ensuring that (1) has a unique solution within $\boldsymbol{\theta} \in \boldsymbol{\Xi}_u$. We also presume that $\cap_u \boldsymbol{\Xi}_u \supset \mathbb{B}(\mathbf{0}_p; r_0) \supset \{\boldsymbol{\theta}_u^*\}_u$, where r_0 is some positive constant. This additional assumption allows us to identify target parameters without disclosing any information about distribution heterogeneity.

With the identifiability of $\boldsymbol{\theta}_u^*$, we need only focus on the heterogeneity that stems from different $\boldsymbol{\theta}_u^*$, which can be described by the adjacency structure among devices.

Definition 1 (Characteristic graph)**.** A graph $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E}_0)$ is the characteristic graph of a set of probability distributions $\{P_u; u \in \mathcal{V}\}$ if $\boldsymbol{\theta}_u^* = \boldsymbol{\theta}_v^*$ is equivalent to $(u, v) \in \mathcal{E}_0$, where $\boldsymbol{\theta}_u^*$ is defined by (1) for all u.

We also refer to $\mathcal{G}_0$ as the characteristic graph of a set of parameters $\{\boldsymbol{\theta}_u^*; u \in \mathcal{V}\}$. By definition, $\mathcal{G}_0$ is composed of multiple disjoint cliques $\mathcal{C}_k$, $k = 1, \ldots, K(\mathcal{G}_0)$, and each clique has its own unique target parameter. Depending on the number of cliques, our model can span across a broad spectrum, from homogeneous distributed learning to multi-task learning setups. For instance, when $K(\mathcal{G}_0) = 1$, it indicates that all device parameters are equivalent, in which case our model aligns with the homogeneous distributed learning setup considered in Stich (2019). Conversely, when $K(\mathcal{G}_0) = V$, it implies that all device parameters are unique, leading us to the multi-task learning setup described by Smith et al. (2017). The number of cliques $K(\mathcal{G}_0)$ quantifies the degree of heterogeneity.

We can alternatively represent heterogeneous target parameters using piecewise constant functions (Fan and Guan 2018; Gao, Han, and Zhang 2020). To see this, define the set consisting of matrices whose rows, after some enumeration, are piecewise constant with at most K pieces,

$$\Xi_p(V, K, \pi) = \left\{ \begin{array}{c} (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_V)^T \in \mathbb{R}^{V \times p} : \text{ there exist sequences } \{a_j\}_{j=0}^K \text{ and } \{\boldsymbol{\mu}_j\}_{j=1}^K \text{ such} \\ \text{that } 0 = a_0 \leq \cdots \leq a_K = V \text{ and } \boldsymbol{\theta}_{\pi(i)} = \boldsymbol{\mu}_j \text{ for integers } i \in (a_{j-1}, a_j] \end{array} \right\},$$

where π is the enumeration mapping, incorporated to aggregate pieces having the same value. Noting that π can be arbitrary, the target parameter set, composed by all matrices whose rows derive their values from at most K distinct vectors, is $\cup_\pi \Xi_p(V, K, \pi)$.

When the enumeration mapping π is known in advance, the minimax rate for estimating parameters in $\Xi_1(V, K, \pi)$ has the order $O\{(nV)^{-1} K \log(eV/K)\}$ (Fan and Guan 2018). Ignoring the logarithmic factor, the rate is the same as the best possible rate corresponding to the case where the characteristic graph $\mathcal{G}_0$ is known. This result is not a surprise since π encodes rich prior information of the distribution heterogeneity; it is much likely for the devices whose indexes are adjacent in π to have the same target parameter.

In practice, unfortunately, a priori we neither know the characteristic graph nor the mapping π. A natural question is whether there exist algorithms that can adapt to the adjacency structure of $\mathcal{G}_0$. The following proposition gives a negative answer that adaptation is impossible when we only know the number of cliques in $\mathcal{G}_0$.

Proposition 1. *Suppose that on the device u we observe $z_1^{(u)}, \dots, z_n^{(u)}$ independently from the Gaussian distribution $N(\theta_u^*, \sigma^2)$ for all u. Then for $K \geq 3$ and $V \geq 5$, there exists some universal constant $\kappa > 0$ such that*

$$\inf_{\widehat{\theta}_u, u \in \mathcal{V}} \sup_{\boldsymbol{\theta}^* \in \cup_\pi \boldsymbol{\Xi}_1(V,K,\pi)} \frac{1}{V} \mathbb{E} \sum_{u \in \mathcal{V}} \left(\widehat{\theta}_u - \theta_u^*\right)^2 \geq \kappa \frac{\sigma^2}{n}, \tag{2}$$

where $\boldsymbol{\theta}^* = (\theta_1^*, \dots, \theta_V^*)^T$ and the infinum is taken over all measurable functions of $\{z_i^{(u)}\}_{u,i}$.

The risk bound $O(\sigma^2/n)$ is achieved by the local sample mean $(1/n)\sum_{i=1}^n (z_i^{(1)}, \dots, z_i^{(V)})^T$, indicating that the lower bound is sharp in the minimax sense. It also reveals that any estimator will be no significantly better than the local estimator, and thus adaptation is impossible unless additional knowledge of $\mathcal{G}_0$ is provided.

In this work, we postulate the existence of an alternative graph, denoted by $\mathcal{G}$, serving as a surrogate for the characteristic graph $\mathcal{G}_0$. Specifically, the underlying assumption is that edges in $\mathcal{G}$ are expected to be found in $\mathcal{G}_0$, albeit with potential discrepancies. This setup includes the case where the enumeration mapping π is known. To illustrate, a given π corresponds to a graph $\mathcal{G}_\pi = (\mathcal{V}, \mathcal{E}_\pi)$, where $(i,j) \in \mathcal{E}_\pi$ if and only if $|\pi(i) - \pi(j)| = 1$. By construction, $\mathcal{G}_\pi$ forms a linear chain comprising $|\mathcal{E}_\pi| = V - 1$ edges, and the target parameters remain piecewise constant with $K(\mathcal{G}_0)$ pieces along this chain. Most edges in $\mathcal{G}_\pi$ also exist in $\mathcal{G}_0$, save for $K(\mathcal{G}_0) - 1$ edges which connect devices belonging to disparate pieces. Indeed, it is important to highlight that $\mathcal{G}$ can be an arbitrary graph, provided it retains some knowledge relevant to $\mathcal{G}_0$. In subsequent sections, we delve into leveraging the prior information in $\mathcal{G}$ for enhanced parameter estimation.

3 Methodology

If there is no heterogeneity, we can aggregate all available samples to estimate a common parameter for all devices, which is referred to as the global estimator,

$$\widehat{\boldsymbol{\theta}}^{\mathrm{gl}} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \cap_u \boldsymbol{\Xi}_u} \sum_{u \in \mathcal{V}} \widehat{M}_u(\boldsymbol{\theta}), \tag{3}$$

where $\widehat{M}_u(\boldsymbol{\theta}) = n^{-1} \sum_{k=1}^{n} m_u(\mathbf{z}_k^{(u)}; \boldsymbol{\theta})$ denotes the empirical risk function on device u. In the presence of heterogeneity, however, the global estimator is inconsistent. A conservative approach for estimating the target parameter on the device u is to exclusively use its own samples, which we term the local estimator,

$$\widehat{\boldsymbol{\theta}}_u^{\mathrm{loc}} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Xi}_u} \widehat{M}_u(\boldsymbol{\theta}), \quad u \in \mathcal{V}. \tag{4}$$

Although the local estimator is asymptotic normal under regularity conditions, it fails to use additional identically distributed samples stored on other devices. This oversight could lead to a significant loss in statistical power (Wolfson et al. 2010; Dobriban and Sheng 2021).

In this paper, it is posited that devices connected in $\mathcal{G}$ are likely to share the same target parameter. To leverage this prior knowledge, we introduce a network-fusion penalty, designed to produce closer estimates for connected devices. Specifically, we propose a network-fusion penalized M-estimator defined by

$$\widehat{\boldsymbol{\Theta}} = \operatorname*{argmin}_{\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_V)^T} F_\lambda(\boldsymbol{\Theta}) \equiv \sum_{u \in \mathcal{V}} \widehat{M}_u(\boldsymbol{\theta}_u) + \lambda R(\mathbf{D}\boldsymbol{\Theta}), \tag{5}$$

where $\mathbf{D}\boldsymbol{\Theta} = (\boldsymbol{\theta}_{e^+} - \boldsymbol{\theta}_{e^-} : e \in \mathcal{E})^T \in \mathbb{R}^{E \times p}$, $R(\mathbf{D}\boldsymbol{\Theta}) = \sum_{e \in \mathcal{E}} \phi(\boldsymbol{\theta}_{e^+} - \boldsymbol{\theta}_{e^-})$ with $\phi(\cdot)$ being a norm on $\mathbb{R}^p$ such as $\|\cdot\|_2$, and λ is a tuning parameter. The objective function $F_\lambda(\boldsymbol{\Theta})$ consists of two terms, the data fidelity term $\sum_{u \in \mathcal{V}} \widehat{M}_u(\boldsymbol{\theta}_u)$ that estimates target parameters per device, and the regularization term $\lambda R(\mathbf{D}\boldsymbol{\Theta})$ that drives the discrepancy between $\boldsymbol{\theta}_{e^+}$ and $\boldsymbol{\theta}_{e^-}$ towards zero for all $e \in \mathcal{E}$.

Notably, the value of λ reflects the extent to which we trust the prior information encapsulated in $\mathcal{G}$. When $\lambda = 0$, the surrogate graph $\mathcal{G}$ has no impact. In this case, since the data fidelity term is separable across u, our proposed estimator coincides with the local estimator in (4). Conversely, for a sufficiently large λ, the regularization term essentially imposes a constrain $\boldsymbol{\theta}_i^* = \boldsymbol{\theta}_j^*$ for all $(i,j) \in \mathcal{E}$. In particular, when $\mathcal{G}$ is connected and λ is large enough, our proposed estimator will behave similarly to the global estimator in (3). The performance of the network fusion penalized estimator relies crucially on the choice of λ. We will provide a general statistical convergence guarantee of $\widehat{\boldsymbol{\Theta}}$ so that the optimal scaling of λ can be determined.

Subsequently, we operate under a regularity assumption that $m_u(\mathbf{z};\boldsymbol{\theta})$ is twice differentiable with respect to $\boldsymbol{\theta}$ almost surely under the probability measure P_u for all u. We write the score function for the device u as $\boldsymbol{\psi}_u(\mathbf{z};\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} m_u(\mathbf{z};\boldsymbol{\theta})$ and the empirical Hessian matrix as $\widehat{\mathbf{H}}_u(\boldsymbol{\theta})$.

3.1 Theoretical Properties

Owing to the rank deficiency of the incidence matrix $\mathbf{D}$, the network-fusion penalized estimator $\widehat{\boldsymbol{\Theta}}$ suffers from non-uniqueness issues. Specifically, given estimators $\widehat{\boldsymbol{\theta}}_i$ and $\widehat{\boldsymbol{\theta}}_j$ for any $(i,j) \in \mathcal{E} \cap \mathcal{E}_0$, there may exist a common shift in both estimators without changing the value of the objective function. To rule out this situation, we directly posit the strong convexity of empirical risk functions.

Condition 2. For some constant $\kappa \geq 1$, we have

$$\kappa^{-1} \leq \min_{u \in \mathcal{V}} \inf_{\boldsymbol{\theta} \in \boldsymbol{\Xi}_u} \lambda_{\min}\big\{\widehat{\mathbf{H}}_u(\boldsymbol{\theta})\big\} \leq \max_{u \in \mathcal{V}} \sup_{\boldsymbol{\theta} \in \boldsymbol{\Xi}_u} \lambda_{\max}\big\{\widehat{\mathbf{H}}_u(\boldsymbol{\theta})\big\} \leq \kappa.$$

Condition 2 frequently appears in studies on the convergence of algorithms, particularly in the context of federated learning (Stich 2019; Li et al. 2020). In our statistical analysis, since the dimension p of target parameters is fixed, this requirement can be easily satisfied; under mild conditions on P_u, we show in Lemma S.1 that Condition 2 holds with high probability.

Another aspect that complicates the provision of statistical guarantees for $\widehat{\boldsymbol{\Theta}}$ is the issue of high-dimensionality; that is, the number of edges in $\mathcal{G}$ is much larger than the local sample size n. In fact, our proposed method in (5) bears some resemblance to the high-dimensional M-estimation penalized by the group Lasso. For better illustration, suppose that $\phi(\cdot) = \|\cdot\|_2$ and $m_u(\mathbf{z}^{(u)}; \boldsymbol{\theta}_u) = \|\mathbf{z}^{(u)} - \boldsymbol{\theta}_u\|_2^2$ for all u. If we further assume that $\mathbf{D}$ has a left inverse $\mathbf{D}^\dagger$ so that $\mathbf{D}^\dagger\mathbf{D} = \mathbf{I}_V$, then our proposed estimator $\widehat{\boldsymbol{\Theta}}$ has the representation $\widehat{\boldsymbol{\Theta}} = \mathbf{D}^\dagger\widehat{\boldsymbol{\Delta}}$, where

$$\widehat{\boldsymbol{\Delta}} \equiv \operatorname*{argmin}_{\boldsymbol{\Delta}=\mathbf{D}\boldsymbol{\Theta}:\boldsymbol{\Theta}\in\mathbb{R}^{V\times p}} \frac{1}{n}\sum_{u\in\mathcal{V}}\sum_{k=1}^{n} \|\mathbf{z}_k^{(u)} - (\mathbf{D}^\dagger\boldsymbol{\Delta})_{u,:}\|_2^2 + \lambda\sum_{e\in\mathcal{E}}\|\boldsymbol{\Delta}_{e,:}\|_2 \tag{6}$$

and $\mathbf{A}_{i,:}$ denotes the ith row of the matrix $\mathbf{A}$. The estimator $\widehat{\boldsymbol{\Delta}}$ is exactly an M-estimator penalized by the group Lasso, where the number of edges in $\mathcal{G}$ is the number of groups and $\mathbf{D}^\dagger$ serves as the design matrix at each group. In high-dimensional settings, some regularity condition of the design matrix is required for parameter identification, for example, the compatibility condition (Bühlmann and Van De Geer 2011). Motivated by the reformulation (6), in our case a similar regularity condition is imposed on $\mathbf{D}$. We first define the set of restrictions by $\mathbb{C}(\mathcal{T}, L) = \{\boldsymbol{\Delta} : R(\boldsymbol{\Delta}_{\mathcal{T}^c,:}) \leq LR(\boldsymbol{\Delta}_{\mathcal{T},:}) \neq 0\}$ for some positive constant L.

Condition 3 (Compatibility factor)**.** The compatibility factor of $\mathbf{D}$ for the set $\mathcal{S} = \mathcal{E} \setminus \mathcal{E}_0$ with respect to $R(\cdot)$ is bounded from below; that is, $\kappa_{\mathcal{S}}(\mathbf{D}) \geq \kappa_0$, where κ_0 is a positive constant,

$$\kappa_{\mathcal{S}}(\mathbf{D}) = \inf_{\boldsymbol{\Theta}:\mathbf{D}\boldsymbol{\Theta}\in\mathbb{C}(\mathcal{S},3)} \frac{\sqrt{|\mathcal{S}|}\|\boldsymbol{\Theta}\|_F}{R\big((\mathbf{D}\boldsymbol{\Theta})_{\mathcal{S},:}\big)},$$

and $(\mathbf{D}\boldsymbol{\Theta})_{\mathcal{S},:}$ denotes the submatrix of $\mathbf{D}\boldsymbol{\Theta}$ with rows indexed by $\mathcal{S}$.

It is worth noting that in general

$$\inf_{\boldsymbol{\Theta}:\mathbf{D}\boldsymbol{\Theta}\in\mathbb{C}(\mathcal{S},3)} \frac{\sqrt{|\mathcal{S}|}\|\boldsymbol{\Theta}\|_F}{R[(\mathbf{D}\boldsymbol{\Theta})_{\mathcal{S},:}]} \geq \inf_{\boldsymbol{\Delta}\in\mathbb{C}(\mathcal{S},3)} \frac{\sqrt{|\mathcal{S}|}\|\mathbf{D}^\dagger\boldsymbol{\Delta}\|_F}{R(\boldsymbol{\Delta}_{\mathcal{S},:})},$$

and thus Condition 3 is slightly weaker than the compatibility condition for the group Lasso problem (6). Our definition of the compatibility factor is a generalization of the notion defined in Hütter and Rigollet (2016) which is specifically designed for $p = 1$ and $\phi(\cdot) = \|\cdot\|_1$ and take the infimum over all $\boldsymbol{\Theta} \in \mathbb{R}^V$. Moreover, it directly follows from Lemma 3 of Hütter and Rigollet (2016) that $\kappa_{\mathcal{S}}(\mathbf{D}) \geq 1/(2\sqrt{d})$, where d denotes the maximum degree of $\mathcal{G}$. Therefore, for graphs with a bounded maximal degree, Condition 3 is satisfied. We assert a deterministic statement about the regularized estimator $\widehat{\boldsymbol{\Theta}}$ that is applicable for any distributions $\{P_u\}_u$ and surrogate graph $\mathcal{G}$, provided the previously mentioned conditions are satisfied. For convenience, let $\widehat{\boldsymbol{\Psi}}(\boldsymbol{\Theta}) = \big(\nabla_{\boldsymbol{\theta}_1}\widehat{M}_1(\boldsymbol{\theta}_1), \ldots, \nabla_{\boldsymbol{\theta}_V}\widehat{M}_V(\boldsymbol{\theta}_V)\big)^T$ and $R^*(\cdot)$ the dual norm of $R(\cdot)$ defined in Lemma S.4. Let $\mathrm{Ker}(\mathbf{D}) = \{\mathbf{u} \in \mathbb{R}^V : \mathbf{D}\mathbf{u} = \mathbf{0}_E\}$ be the kernel space of $\mathbf{D}$. We also denote by $\boldsymbol{\Pi}_{\mathrm{Ker}(\mathbf{D})}$ the projection matrix that maps vectors in $\mathbb{R}^V$ to the kernel space of $\mathbf{D}$. Define $S = |\mathcal{E} \setminus \mathcal{E}_0|$ and $\rho = \|\boldsymbol{\Pi}_{\mathrm{Ker}(\mathbf{D})}\widehat{\boldsymbol{\Psi}}(\boldsymbol{\Theta}^*)\|_F$.

Theorem 2. *Under the Conditions 1–3, the network-fusion penalized estimator with* $\lambda = 2R^*\big\{(\mathbf{D}^\dagger)^T\widehat{\boldsymbol{\Psi}}(\boldsymbol{\Theta}^*)\big\}$ *satisfies that*

$$\big\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\big\|_F \leq \kappa\bigg(\rho + \frac{2\sqrt{S}}{\kappa_0}\lambda\bigg). \tag{7}$$

Our deterministic bound includes two components, ρ^2 and $S\lambda^2/\kappa_0$, which separately controls the error for estimating $\boldsymbol{\Pi}_{\mathrm{Ker}(\mathbf{D})}\boldsymbol{\Theta}^*$ and $(\mathbf{I}_V - \boldsymbol{\Pi}_{\mathrm{Ker}(\mathbf{D})})\boldsymbol{\Theta}^*$. To see this, by Theorem 8.3.1 of Godsil and Royle (2001) we have $\mathrm{Ker}(\mathbf{D}) = \mathrm{span}\{\mathbf{1}_{\mathcal{C}_1}, \ldots, \mathbf{1}_{\mathcal{C}_{K(\mathcal{G})}}\}$, where $\mathbf{1}_{\mathcal{C}_i} = \big(I(1 \in \mathcal{C}_i), \ldots, I(V \in \mathcal{C}_i)\big)^T$ is the indicator vector of $\mathcal{C}_i$. Thus, ρ^2 can be construed as the *averaged intra-group variances* for estimating rows of $\boldsymbol{\Pi}_{\mathrm{Ker}(\mathbf{D})}\boldsymbol{\Theta}^*$; that is, $|\mathcal{C}_i|^{-1}\mathbf{1}_{\mathcal{C}_i}^T\boldsymbol{\Theta}^*$, $i = 1, \ldots, K(\mathcal{G})$. Furthermore, by (6), our network-fusion regularization mirrors the group Lasso in estimating $(\mathbf{I}_V - \boldsymbol{\Pi}_{\mathrm{Ker}(\mathbf{D})})\boldsymbol{\Theta}^*$. As expected, the rate structure of $S\lambda^2/\kappa_0$ bears a strong resemblance to that of conventional Lasso estimators.

In order to elucidate the relationship between the estimation error and factors such as the local sample size n, the graph size V, and the graph structure $\mathcal{G}$, we need an additional condition on distributions $\{P_u\}_u$ that enables us to derive an explicit representation for ρ and λ.

Condition 4. The random vector $\boldsymbol{\psi}_u(\mathbf{z}_k^{(u)}; \boldsymbol{\theta}_u^*)$ is sub-Gaussian with parameter $\sigma^2 < \infty$ for all u; that is, for any $\mathbf{a} \in \mathbb{R}^p$ with $\|\mathbf{a}\|_2 = 1$,

$$\mathbb{E}\exp\big[\big\{\mathbf{a}^T\boldsymbol{\psi}_u(\mathbf{z}_k^{(u)}; \boldsymbol{\theta}_u^*)\big\}^2/\sigma^2\big] \leq 2, \quad k = 1, \ldots, n.$$

Condition 4 imposes a sub-Gaussian tail on the distribution of noises for technical convenience, which can be relaxed to distributions of high-order moments. Equipped with this condition, we derive the risk bound for $\widehat{\mathbf{\Theta}}$ explicitly depending on n, V, and graph invariants.

Theorem 3. *Under Conditions 1–4, if we choose $\phi(\cdot) = \|\cdot\|_1$ or $\phi(\cdot) = \|\cdot\|_2$, then with probability at least $1-2\xi$ we have*

$$\frac{1}{V}\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}^*\|_F^2 \leq C\sigma^2\left\{\frac{pK(\mathcal{G})\log(1/\xi)}{nV} + \frac{\gamma_{\mathcal{G}}^2}{\kappa_0^2}\frac{pS\log(V/\xi)}{nV}\right\}$$

for some positive constant C, where $\gamma_{\mathcal{G}}$ is the maximum Euclidean norm among all columns of $\mathbf{D}^\dagger$.

The term $\gamma_{\mathcal{G}}^2$ appears when we explicitly bound λ. This quantity, which is invariant under graph isomorphisms, measures the effectiveness of communication between nodes within the graph. When $\mathcal{G}$ is connected, $\gamma_{\mathcal{G}}^2$ is bounded by the inverse of algebraic connectivity of $\mathcal{G}$, which signifies the robustness and synchronizability of the graph (Wu et al. 2011). Notably, for a multitude of graph types, the magnitude of $\gamma_{\mathcal{G}}^2/\kappa_0$ does not grow as the graph size V tends to infinity. For example, Proposition 6 of Hütter and Rigollet (2016) stipulates that $\gamma_{\mathcal{G}}^2/\kappa_0 = O(1)$ for 3-D lattice graphs, as well as their higher-dimensional counterparts. This constant bound also holds for expander graphs with a bounded degree as evidenced by Proposition 1.82 and Corollary 1.87 in Krebs and Shaheen (2011).

Considering $\gamma_{\mathcal{G}}^2/\kappa_0$ as a constant that does not depend on n and V, the convergence rate of our estimator is of order $O_P\big[\sigma^2 p\{K(\mathcal{G}) + S\}(\log V)/(nV)\big]$. Interestingly, lower values of $K(\mathcal{G})$, corresponding to a higher degree of aggregation, are usually associated with higher values of S. This is because additional edges, absent from $\mathcal{G}_0$, might be incorporated into $\mathcal{G}$ to achieve a smaller $K(\mathcal{G})$, demonstrating the inherent trade-off between aggregation and heterogeneity.

In cases where the characteristic graph $\mathcal{G}_0$ is known, the optimal estimator converges at the rate of $O_P\{\sigma^2 pK(\mathcal{G}_0)/(nV)\}$. To assess how well $\mathcal{G}$ serves as a surrogate for $\mathcal{G}_0$, we introduce the concept of *graph fidelity*, which is defined by contrasting two rates,

$$\mathrm{GF}_{\mathcal{G}_0}(\mathcal{G}) \equiv \frac{K(\mathcal{G}_0)}{K(\mathcal{G}) + |\mathcal{E} \setminus \mathcal{E}_0|}.$$

As the graph size increases, surrogate graphs with a non-vanishing graph fidelity will yield a network-fusion regularized estimator whose convergence rate matches the order of the optimal estimator, except for a logarithmic prefactor. This implies that, as $V/K(\mathcal{G}_0)$ becomes larger, our method has the capability to handle a large number of heterogeneous devices simultaneously.

3.2 Edge Selection by Multiple Testing

In this section, our aim is to optimally utilize the prior information encapsulated in $\mathcal{G}$. Specifically, we propose identifying a subgraph of $\mathcal{G}$ that maintains the greatest graph fidelity,

$$\widehat{\mathcal{E}} = \underset{\widetilde{\mathcal{E}} \subset \mathcal{E}}{\operatorname{argmin}} \big\{ K(\widetilde{\mathcal{G}}) + |\widetilde{\mathcal{E}} \setminus \mathcal{E}_0| \big\}, \tag{8}$$

where any subgraph of $\mathcal{G}$ is denoted as $\widetilde{\mathcal{G}} = (\mathcal{V}, \widetilde{\mathcal{E}})$. Since the upper bound of the convergence rate in Theorem 3 is minimized, $\widehat{\mathcal{G}} = (\mathcal{V}, \widehat{\mathcal{E}})$ represents the graph yielding the best network-fusion regularized estimator based on $\mathcal{G}$. When $\mathcal{G}$ contains multiple connected components, the objective function in (8) is separable according to these components. Without loss of generality, we assume that $\mathcal{G}$ is connected such that $K(\mathcal{G}) = 1$. The following proposition shows the connection between problem (8) and the selection of true edges.

Proposition 4. *For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $K(\mathcal{G}) = 1$, we have $\min_{\widetilde{\mathcal{E}} \subset \mathcal{E}} \big\{ K(\widetilde{\mathcal{G}}) + |\widetilde{\mathcal{E}} \setminus \mathcal{E}_0| \big\} = K(\mathcal{G} \cap \mathcal{G}_0)$, where $\mathcal{G} \cap \mathcal{G}_0 = (V, \mathcal{E} \cap \mathcal{E}_0)$.*

Proposition 4 suggests that $\mathcal{E} \cap \mathcal{E}_0$ is one of the solutions for (8). By Definition 1, $(i,j) \in \mathcal{E}_0$ is equivalent to $\boldsymbol{\theta}_i^* = \boldsymbol{\theta}_j^*$. This leads us to perform a simultaneous test of the following hypotheses

$$H_{0,e} \colon \boldsymbol{\theta}_{e^+}^* = \boldsymbol{\theta}_{e^-}^* \text{ versus } H_{1,e} \colon \boldsymbol{\theta}_{e^+}^* \neq \boldsymbol{\theta}_{e^-}^* \tag{9}$$

for all $e \in \mathcal{E}$. We impose Condition 5 for technical convenience.

Condition 5. There exist positive definite matrices $\widehat{\boldsymbol{\Omega}}_u$ such that $\sqrt{n}\widehat{\boldsymbol{\Omega}}_u^{-1/2}(\widehat{\boldsymbol{\theta}}_u^{\text{loc}} - \boldsymbol{\theta}_u^*) \to_d N(\mathbf{0}_p, \mathbf{I}_p)$ for all u.

As demonstrated in Proposition S.3, under additional regularity conditions on $m_u(\mathbf{z};\cdot)$ and P_u, Condition 5 is satisfied with $\widehat{\boldsymbol{\Omega}}_u = \{\widehat{\mathbf{H}}_u(\widehat{\boldsymbol{\theta}}_u^{\text{loc}})\}^{-1}\widehat{\boldsymbol{\Sigma}}_u(\widehat{\boldsymbol{\theta}}_u^{\text{loc}})\{\widehat{\mathbf{H}}_u(\widehat{\boldsymbol{\theta}}_u^{\text{loc}})\}^{-T}$, where $\widehat{\boldsymbol{\Sigma}}_u(\boldsymbol{\theta}) = n^{-1}\sum_i \boldsymbol{\psi}_u(\mathbf{z}_i^{(u)};\boldsymbol{\theta})\{\boldsymbol{\psi}_u(\mathbf{z}_i^{(u)};\boldsymbol{\theta})\}^T$ and $\widehat{\mathbf{H}}_u(\boldsymbol{\theta}) = n^{-1}\sum_i \nabla_{\boldsymbol{\theta}}\boldsymbol{\psi}_u(\boldsymbol{\theta})$. Thus, for $\boldsymbol{\theta}_{e^+}^* = \boldsymbol{\theta}_{e^-}^*$, we can construct a test statistic by

$$\widehat{W}_e = \big\{n\big(\widehat{\boldsymbol{\theta}}_{e^+}^{\text{loc}} - \widehat{\boldsymbol{\theta}}_{e^-}^{\text{loc}}\big)^T\big(\widehat{\boldsymbol{\Omega}}_{e^+} + \widehat{\boldsymbol{\Omega}}_{e^-}\big)^{-1}\big(\widehat{\boldsymbol{\theta}}_{e^+}^{\text{loc}} - \widehat{\boldsymbol{\theta}}_{e^-}^{\text{loc}}\big)\big\}^{1/2}. \tag{10}$$

Adopting the Bonferroni correction, we select $\mathcal{E} \cap \mathcal{E}_0$ by

$$\widehat{\mathcal{E}} = \big\{e \in \mathcal{E} : |\widehat{W}_e|^2 \le \chi_p^2(\alpha/E)\big\}, \tag{11}$$

where $\chi_p^2(\alpha)$ is the upper α-quantile of the χ_p^2 distribution. For $\boldsymbol{\theta}_{e^+}^* \ne \boldsymbol{\theta}_{e^-}^*$, to adaptively measure the distance between them, define the distance $\text{dist}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \big\{\big(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\big)^T\big(\boldsymbol{\Omega}_{e^+}^* + \boldsymbol{\Omega}_{e^-}^*\big)^{-1}\big(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\big)\big\}^{1/2}$, where $\boldsymbol{\Omega}_u^*$ is the probabilistic limit of $\widehat{\boldsymbol{\Omega}}_u^*$ for all u. Under certain minimum signal condition, we show that our procedure can consistently select edges in E_0 with a large probability.

Theorem 5. *Under Condition 5, if further*

$$\min_{e \in \mathcal{E}\setminus\mathcal{E}_0} n\big\{\text{dist}(\boldsymbol{\theta}_{e^+}^*, \boldsymbol{\theta}_{e^-}^*)\big\}^2 \ge 4\chi_p^2(\alpha/E), \tag{12}$$

then $\liminf_{n\to\infty} P\left(\widehat{\mathcal{E}} = \mathcal{E} \cap \mathcal{E}_0\right) \ge 1 - \alpha.$

Theorem 5 illustrates the effectiveness of our selection procedure (11), which incurs neither false negatives nor false positives with a confidence level of $1 - \alpha$. Intriguingly, this procedure operates without the need for inter-device data exchange.

Although this edge selection procedure aims at selecting edges in $\mathcal{G}_0$, our findings from (8) and Proposition 4 reveal an interesting point: the graph that optimizes the convergence rate is not necessarily $\mathcal{G}_0$. In fact, any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, provided it satisfies $K(\mathcal{E} \cap \mathcal{E}_0) = K(\mathcal{G}_0)$, can serve this purpose. Meanwhile, to ensure that $\gamma_{\mathcal{G}}^2/\kappa_0$ remains a bounded constant after edge selection, we can designate $\mathcal{G}$ as any expander graph. This underlines the robustness of our method against graph misspecification.

Nonetheless, (11) is known to be conservative for controlling the false positive rate (FPR) (Benjamini and Hochberg 1995). Despite that we have obtained the asymptotic distribution of the test statistic in (10), controlling the FPR in our edge selection procedure presents a challenge due to dependencies of those test statistics. We recognize this issue as an area ripe for future exploration.

4 Optimization

In this section, we turn our attention to the derivation of the network-fusion regularized estimator. We introduce FedADMM, a decentralized, stochastic version of ADMM, designed to solve the optimization problem described by (5). Unlike traditional methods, FedADMM utilizes only a mini-batch of samples in each iteration per device, and does not require the transmission of local data. Moreover, this algorithm allows for heterogeneous device availability patterns. For simplicity, we assume that edges in $\mathcal{G}$ point from larger nodes to smaller ones, meaning that $(i,j) \in \mathcal{E}$ implies $i > j$. We denote the neighbors of node i as $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}\} \cup \{j : (j,i) \in \mathcal{E}\}$.

4.1 FedADMM

We first consider the case where all devices are available instantaneously. Similar to Hallac, Leskovec, and Boyd (2015), we introduce auxiliary vectors $\boldsymbol{\beta}_{ij}, \boldsymbol{\beta}_{ji}$ subject to the constraints $\boldsymbol{\beta}_{ij} = \boldsymbol{\theta}_i, \boldsymbol{\beta}_{ji} = \boldsymbol{\theta}_j$ for all $(i,j) \in \mathcal{E}$. The resulting augmented Lagrangian (Hestenes 1969) is given by

$$\begin{aligned} L(\boldsymbol{\Theta}, \mathbf{B}, \mathbf{A}) = \frac{1}{V} \sum_{i \in \mathcal{V}} \widehat{M}_i(\boldsymbol{\theta}_i) + \lambda \sum_{(i,j)\in\mathcal{E}} \phi(\boldsymbol{\beta}_{ij} - \boldsymbol{\beta}_{ji}) \\ - \sum_{(i,j)\in\mathcal{E}} \left\{ \boldsymbol{\alpha}_{ij}^T(\boldsymbol{\theta}_i - \boldsymbol{\beta}_{ij}) + \boldsymbol{\alpha}_{ji}^T(\boldsymbol{\theta}_j - \boldsymbol{\beta}_{ji}) \right\} + \frac{\rho}{2} \sum_{(i,j)\in\mathcal{E}} (\|\boldsymbol{\theta}_i - \boldsymbol{\beta}_{ij}\|_2^2 + \|\boldsymbol{\theta}_j - \boldsymbol{\beta}_{ji}\|_2^2), \end{aligned} \tag{13}$$

where $\mathbf{B} = (\boldsymbol{\beta}_{ij}, \boldsymbol{\beta}_{ji} : (i,j) \in \mathcal{E})$ and $\mathbf{A} = (\boldsymbol{\alpha}_{ij}, \boldsymbol{\alpha}_{ji} : (i,j) \in \mathcal{E})$. ADMM typically solves (13) by iteratively minimizing $L(\boldsymbol{\Theta}, \mathbf{B}, \mathbf{A})$ with respect to $\boldsymbol{\Theta}$ and $\mathbf{B}$, while keeping the other fixed, followed by an update of the Lagrangian multiplier $\mathbf{A}$. Importantly, $L(\boldsymbol{\Theta}, \mathbf{B}, \mathbf{A})$ is separable, which allows for the distributed execution of updates for $\boldsymbol{\Theta}, \mathbf{B}$, and $\mathbf{A}$.

In practical settings, local devices often lack the computational resources to optimize with the full dataset. Drawing inspiration from stochastic gradient descent, we modify the approach to optimize $L(\boldsymbol{\Theta}, \mathbf{B}(t), \mathbf{A}(t))$ with respect to $\boldsymbol{\theta}_i$ on device i. Rather than direct minimization, we implement a one-step stochastic gradient update during the t-th iteration:

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) - \eta(t)\Big\{\mathbf{g}_i(t) + \rho \sum_{j \in \mathcal{N}_i} \big(\boldsymbol{\theta}_i(t) - \boldsymbol{\beta}_{ij}(t) - \rho^{-1}\boldsymbol{\alpha}_{ij}(t)\big)\Big\}, \tag{14}$$

where $\eta(t)$ is the learning rate, $\mathcal{B}_i(t)$ represents a mini-batch randomly drawn from $\{\mathbf{z}_k^{(i)}\}_{k=1}^{n_i}$ on device i in the t-th iteration, and $\mathbf{g}_i(t) = |\mathcal{B}_i(t)|^{-1} \sum_{b \in \mathcal{B}_i(t)} \boldsymbol{\psi}_i(\mathbf{z}_b^{(i)}; \boldsymbol{\theta}_i(t))$ is an unbiased estimator of $\nabla_{\boldsymbol{\theta}} \widehat{M}_i(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_i(t)$. Besides local samples, the update equation (14) only requires $\boldsymbol{\beta}_{ij}(t)$ and $\boldsymbol{\alpha}_{ij}(t)$, both of which can be sourced from device j. This enables the simultaneous execution of (14) on all devices.

With $\boldsymbol{\theta}_i(t+1)$ updated, we proceed to update $\boldsymbol{\beta}_{ij}(t)$ and $\boldsymbol{\beta}_{ji}(t)$ using

$$\begin{pmatrix} \boldsymbol{\beta}_{ij}(t+1) \\ \boldsymbol{\beta}_{ji}(t+1) \end{pmatrix} = \underset{\boldsymbol{\beta}_{ij}, \boldsymbol{\beta}_{ji}}{\operatorname{argmin}} \Big\{ \lambda\phi(\boldsymbol{\beta}_{ij} - \boldsymbol{\beta}_{ji}) + \frac{\rho}{2}\|\boldsymbol{\theta}_i(t+1) - \boldsymbol{\beta}_{ij} - \rho^{-1}\boldsymbol{\alpha}_{ij}(t)\|_2^2 \\ + \frac{\rho}{2}\|\boldsymbol{\theta}_j(t+1) - \boldsymbol{\beta}_{ji} - \rho^{-1}\boldsymbol{\alpha}_{ji}(t)\|_2^2 \Big\}. \tag{15}$$

Either device i or j can implement (15), as long as the necessary parameters are transmitted to the correct device. It is worth noting that, for $\phi(\cdot) = \|\cdot\|_1$ and $\phi(\cdot) = \|\cdot\|_2$, we derive an explicit update equation from (15) in Lemma S.11 in Supplementary Materials. Finally, we update $\boldsymbol{\alpha}_{ij}(t)$ and $\boldsymbol{\alpha}_{ji}(t)$ with

$$\begin{pmatrix} \boldsymbol{\alpha}_{ij}(t+1) \\ \boldsymbol{\alpha}_{ji}(t+1) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\alpha}_{ij}(t) \\ \boldsymbol{\alpha}_{ji}(t) \end{pmatrix} - \rho \begin{pmatrix} \boldsymbol{\theta}_i(t+1) - \boldsymbol{\beta}_{ij}(t+1) \\ \boldsymbol{\theta}_j(t+1) - \boldsymbol{\beta}_{ji}(t+1) \end{pmatrix}. \tag{16}$$

Note that update equation (16) , like the previous update equation, only requires parameter exchange among connected devices. Both (15) and (16) can be performed in parallel across all edges. We refer to (14) as the node optimization step, and (15) and (16) as the edge communication step. FedADMM is detailed in Algorithm 1.

```
Algorithm 1: Decentralized stochastic ADMM
  Input: Initial value $\boldsymbol{\Theta}(0), \mathbf{B}(0), \mathbf{A}(0)$, number of iterations $T$, and the learning rate
         $\eta(t), t = 1, \ldots, T$.
  repeat
      Sample mini-batches $\mathcal{B}_i(t)$ on device $i$ in parallel;
      Obtain $\boldsymbol{\theta}_i(t+1)$ on device $i$ by (14) in parallel for each $i \in \mathcal{V}$;
      Broadcast each $\boldsymbol{\theta}_i(t+1)$ to neighbor devices;
      Obtain $\boldsymbol{\beta}_{ij}(t+1)$ and $\boldsymbol{\beta}_{ji}(t+1)$ on device $i$ by (15) in parallel for $(i,j) \in \mathcal{E}$;
      Obtain $\boldsymbol{\alpha}_{ij}(t+1)$ and $\boldsymbol{\alpha}_{ji}(t+1)$ on device $i$ by (16) in parallel for $(i,j) \in \mathcal{E}$;
      Broadcast $(\boldsymbol{\beta}_{ij}(t+1), \boldsymbol{\beta}_{ji}(t+1))$ and $(\boldsymbol{\alpha}_{ij}(t+1), \boldsymbol{\alpha}_{ji}(t+1))$ from device $i$ to
        device $j$ in parallel for $(i,j) \in \mathcal{E}$;
      $t \leftarrow t+1$
  until $t > T$;
  Output: $\overline{\boldsymbol{\Theta}} = T^{-1} \sum_{t=1}^{T} \boldsymbol{\Theta}(t-1)$
```

Our algorithm bears similarity to methods developed in Ouyang et al. (2013) and Suzuki (2013). They approximated $m_i(\cdot; \boldsymbol{\theta}_i)$ with a linear function $m_i(\cdot; \boldsymbol{\theta}_i(t)) + (\boldsymbol{\theta} - \boldsymbol{\theta}_i(t))^T \boldsymbol{\psi}_i(\cdot; \boldsymbol{\theta}_i(t))$, and used the proximal method (Rockafellar 1976) to update $\boldsymbol{\theta}_i(t)$,

$$\boldsymbol{\theta}_i(t+1) = \underset{\boldsymbol{\theta} \in \Xi_i}{\operatorname{argmin}} \left\{ \boldsymbol{\theta}^T \mathbf{g}_i(t) + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \|\boldsymbol{\theta} - \boldsymbol{\beta}_{ij}(t) - \rho^{-1} \boldsymbol{\alpha}_{ij}(t)\|_2^2 + \frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}_i(t)\|_2^2}{2\widetilde{\eta}(t)} \right\}, \tag{17}$$

where $\widetilde{\eta}(t)$ is the step-size. It is worth noting that our update equation for $\boldsymbol{\theta}$ offering an extension of (17) with the learning rate that adapts to the size of the neighboring nodes. Specifically, by setting $\eta(t) = \widetilde{\eta}(t)/(1 + \rho|\mathcal{N}_i|\widetilde{\eta}(t))$, one can verify that (17) is equivalent to (14).

In the sequel, we show that the output of Algorithm (1) converges to the global minimizer of (13). Let $\big\{\boldsymbol{\theta}_i(t), \boldsymbol{\beta}_{ij}(t), \boldsymbol{\beta}_{ji}(t), \boldsymbol{\alpha}_{ij}(t), \boldsymbol{\alpha}_{ji}(t) : i, j \in \mathcal{V}\big\}$ be the output of Algorithm 1 in the t-th iteration for $t = 0, \ldots, T$. Denote by $\big\{\widehat{\boldsymbol{\theta}}_i, \widehat{\boldsymbol{\beta}}_{ij}, \widehat{\boldsymbol{\beta}}_{ji}, \widehat{\boldsymbol{\alpha}}_{ij}, \widehat{\boldsymbol{\alpha}}_{ji} : i, j \in \mathcal{V}\big\}$ the global minimizer of (13). Define the ball $\Xi = \mathbb{B}(\mathbf{0}_p; r_0)$ for some constant r_0 such that $\{\boldsymbol{\theta}_u^*\}_u \subset \Xi \subset \cap_u \Xi_u$. Without loss of generality, we assume that $\big\{\widehat{\boldsymbol{\theta}}_i, \widehat{\boldsymbol{\beta}}_{ij}, \widehat{\boldsymbol{\beta}}_{ji}, \boldsymbol{\theta}_i(t), \boldsymbol{\beta}_{ij}(t), \boldsymbol{\beta}_{ji}(t)$:

$i, j \in \mathcal{V}, 0 \leq t \leq T\} \subset \boldsymbol{\Xi}$, since we can always project them into $\boldsymbol{\Xi}$. For simplicity, let $C_\psi = \max_{u \in \mathcal{V}} \sup_{\boldsymbol{\theta} \in \boldsymbol{\Xi}} n^{-1} \sum_{k=1}^{n} \|\boldsymbol{\psi}_u(\mathbf{z}_k^{(u)}; \boldsymbol{\theta})\|_2^2$ and $\kappa_\alpha = \max_{i \in \mathcal{V}, j \in \mathcal{N}_i}(\|\boldsymbol{\alpha}_{ij}(0)\|_2^2 + \|\widehat{\boldsymbol{\alpha}}_{ij}\|_2^2)$.

Theorem 6. *Under Conditions 1 and 2, if $\kappa_\alpha \geq \lambda \sup_{\mathbf{a} \neq \mathbf{0}} \phi(\mathbf{a}) \|\mathbf{a}\|_2^{-1}$ and $C_\psi < \infty$, then by choosing the learning rate as $\eta(t) = \kappa/t$ we have*

$$\frac{1}{V} \mathbb{E} \|\overline{\boldsymbol{\Theta}} - \widehat{\boldsymbol{\Theta}}\|_F^2 \leq \frac{2\kappa^2 C_\psi \log T}{T},$$

for sufficiently large T such that $\kappa C_\psi V \log T \geq E(8\rho^{-1}\kappa_\alpha + 2r_0^2\rho + 4r_0\kappa_\alpha)$, where the expectation is taken with respect to the choice of mini-batches $\{\mathcal{B}_u(t) : u \in \mathcal{V}, t = 1, \ldots, T\}$.

The conditions on κ_α and C_ψ are not restrictive. Noting that the optimal λ decreases to zero as n or V grows, but $\sup_{\mathbf{a} \neq \mathbf{0}} \phi(\mathbf{a}) \|\mathbf{a}\|_2^{-1}$ does not, a large initial $\boldsymbol{\alpha}$ ensures that $\kappa_\alpha > \lambda \sup_{\mathbf{a} \neq \mathbf{0}} \phi(\mathbf{a}) \|\mathbf{a}\|_2^{-1}$. Additionally, $C_\psi < \infty$ is satisfied when $n^{-1} \sum_k \|\boldsymbol{\psi}_u(\mathbf{z}_k^{(u)}; \boldsymbol{\theta})\|_2^2$ is continuous with respect to $\boldsymbol{\theta}$. Our result shows that for $T > O\{nV/(pK(\mathcal{G}) + pS)\}$, the optimization error introduced by the algorithm becomes less significant than the inherent statistical error.

4.2 Extension of FedADMM to Heterogeneous Accessibility of Devices

In this subsection, we explore a scenario where devices might become inaccessible; that is, they may go offline or become unavailable during the real-time optimization process. For each device $i \in \mathcal{V}$ and iteration $t \in \mathbb{N}$, let $R_i(t) = I(\text{device } i \text{ is available in the iteration } t)$. We denote the set of available devices in the t-th iteration as $\mathcal{S}(t) = \{i : R_i(t) = 1\}$.

The inaccessibility of devices is modeled as a random variable $R_i(t)$ from the Bernoulli distribution with mean p_i. Furthermore, we assume that the process $R_i(t), t > 0$ holds the memoryless property; that is, $\{R_i(t_1) : i \in \mathcal{V}\}$ is independent of $\{R_i(t_2) : i \in \mathcal{V}\}$ for any $t_1 \neq t_2$. We also consider scenarios where the probability of inaccessibility varies among devices; that is, $p_i \neq p_j$ for $i \neq j$. We allow for dependencies among $\{R_i(t) : i \in \mathcal{V}\}$ for a fixed t. Finally, we assume the positivity condition $p_0 \equiv \min_{i \in \mathcal{V}} p_i > 0$.

To begin with, we presume the existence of a central server $\mathcal{O}$ and knowledge of $p_i, i \in \mathcal{V}$. Drawing on the inverse probability weighting method (Wooldridge 2007) from causal inference literature, an unbiased estimator of $\nabla_{\boldsymbol{\theta}_i} \widehat{M}_i(\boldsymbol{\theta}_i(t))$ is given by

$$\widetilde{\mathbf{g}}_i(t) = \frac{1}{|\mathcal{B}_i(t)|} \sum_{b \in \mathcal{B}_i(t)} \frac{R_i(t)}{p_i} \boldsymbol{\psi}_i(\mathbf{z}_b^{(i)}; \boldsymbol{\theta}_i(t)). \tag{18}$$

Therefore, we modify (14) as

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) - \eta(t) \left\{ \widetilde{\mathbf{g}}_i(t) + \rho \sum_{j \in \mathcal{N}_i} (\boldsymbol{\theta}_i(t) - \boldsymbol{\beta}_{ij}(t) - \rho^{-1} \boldsymbol{\alpha}_{ij}(t)) \right\}. \tag{19}$$

If device i is accessible during the iteration t, we need to send $\boldsymbol{\theta}_i(t)$ from $\mathcal{O}$ to device i. We then calculate $\widetilde{\mathbf{g}}_i(t)$ on device i and send $\widetilde{\mathbf{g}}_i(t)$ back to $\mathcal{O}$ to update $\boldsymbol{\theta}_i(t+1)$. If the device is not accessible, by (18) we directly set $\widetilde{\mathbf{g}}_i(t) = \mathbf{0}_p$ in (19). Once $\boldsymbol{\theta}_i(t+1)$ (for $i \in \mathcal{V}$) has been updated, we can calculate $\big(\boldsymbol{\beta}_{ij}(t+1), \boldsymbol{\beta}_{ji}(t+1)\big)$ using (15), and $\big(\boldsymbol{\alpha}_{ij}(t+1), \boldsymbol{\alpha}_{ji}(t+1)\big)$ using (16) on $\mathcal{O}$.

It should be noted that the presence of a central server is optional, as each device can maintain a copy of the parameters from other devices. Specifically, during the t-th iteration, if device i cannot receive $\boldsymbol{\theta}_j(t+1)$ from a neighboring device (for instance, device j), device i can retrieve $\boldsymbol{\theta}_j(t+1)$ using equation (19) with $\widetilde{\mathbf{g}}_i(t) = \mathbf{0}_p$. This is possible provided device i has kept a record of $\boldsymbol{\alpha}_{jk}(t)$ and $\boldsymbol{\beta}_{jk}(t)$ for all $k \in \mathcal{N}_j$. Although this method increases the communication cost, it is still practical. Additionally, it is essentially unnecessary to know $(p_1, \ldots, p_V)^T$ a priori. In the t-th iteration, the vector $(p_1, \ldots, p_V)^T$ can be estimated by the frequency of each device being offline over the first $t-1$ iterations. The FedADMM approach with randomly inaccessible devices is outlined in Algorithm 2. For simplicity, we maintain the central server in this algorithm.

Similar to Theorem 6, we obtain the convergence rate of Algorithm 2 in Supplementary materials.

Algorithm 2: FedADMM with randomly inaccessible devices

```
Input  : Initial value $\mathbf{\Theta}(0), \boldsymbol{\beta}(0), \boldsymbol{\alpha}(0)$, number of iterations $T$, and the learning
         rate $\eta(t), t = 1, \dots, T, i \in \mathcal{V}$.
repeat
    for $i \in \mathcal{S}(t)$ do
        Broadcast $\boldsymbol{\theta}_i(t)$ from $\mathcal{O}$ to device $i$ ;
        Obtain $\widetilde{\mathbf{g}}_i(t)$ by (18) with $p_i = p_i(t)$ in parallel and send $\widetilde{\mathbf{g}}_i(t)$ back to $\mathcal{O}$;
    Obtain $\boldsymbol{\theta}_i(t+1)$ by (19) in $\mathcal{O}$;
    Obtain $\boldsymbol{\beta}_{ij}(t+1)$ and $\boldsymbol{\beta}_{ji}(t+1)$ by (15) for $(i, j) \in \mathcal{E}$ in $\mathcal{O}$;
    Obtain $\boldsymbol{\alpha}_{ij}(t+1)$ and $\boldsymbol{\alpha}_{ji}(t+1)$ by (16) for $(i, j) \in \mathcal{E}$ in $\mathcal{O}$;
    If $(p_1, \dots, p_V)^T$ is unknown, record $(R_i(t) : i \in \mathcal{V})$ and update
     $p_i(t+1) = t^{-1} \sum_{t=1}^{t} R_i(t)$, $i \in \mathcal{V}$;
    $t \leftarrow t + 1$;
until $t > T$;
Output: $\overline{\mathbf{\Theta}} = T^{-1} \sum_{t=1}^{T} \mathbf{\Theta}(t-1)$
```

Corollary 7. *Under the same conditions of Theorem 6, if $(p_1, \dots, p_V)^T$ is known and $p_0 = \min_{i \in \mathcal{V}} p_i > 0$, then the output of Algorithm 2 satisfies that*

$$\frac{1}{V} \mathbb{E} \| \overline{\mathbf{\Theta}} - \widehat{\mathbf{\Theta}} \|_F^2 \leq \frac{2\kappa^2 C_\psi \log T}{p_0 T},$$

for sufficiently large T, where the expectation is taken with respect to the choice of mini-batches $\{\mathcal{B}_u(t) \colon u \in \mathcal{V}, t = 1, \dots, T\}$.

5 Simulations

In this section, we evaluate the performance of six methods in various settings: FedADMM, FedADMM-ES, FedADMM-Local-ES, Oracle, Local, and Global. For a surrogate graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, FedADMM, FedADMM-ES, and FedADMM-Local-ES represent the outputs of Algorithm 1 with $\mathcal{G}$, after applying the adaptive edge selection procedure to $\mathcal{G}$, and after applying the adaptive edge selection procedure to the complete graph respectively. Oracle represents the output of Algorithm 1 with the characteristic graph $\mathcal{G}_0$. Meanwhile, Local and Global correspond to the local estimator and the global estimator defined in (4) and (3), respectively. The performance metric is chosen as the average squared estimation error $\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}^*\|_F^2 / V$. Additionally, we compare the convergence rates of FedADMM and gradient-descent-based methods in solving the objective function (5) in terms of the total number of iterations required for convergence.

We introduce the data generating process of our simulation. We consider linear regression tasks on each device, where covariates $\mathbf{x}_k^{(u)} \sim N_p(0, \mathbf{I}_p)$ and noise variables $\varepsilon_k^{(u)} \sim N(0,1)$ are generated independently for $k = 1, \ldots, n; u \in \mathcal{V}$. The characteristic graph $\mathcal{G}_0$ is created by partitioning $\mathcal{V}$ into K_0 evenly-sized subsets, $\mathcal{V}_1, \ldots, \mathcal{V}_{K_0}$, with each set $\mathcal{V}_j$ forming a complete subgraph. We then store the adjacency matrix $\mathbf{\Lambda}_0$ of $\mathcal{G}_0$. For each subset $\mathcal{V}_j$ and node $u \in \mathcal{V}_j$, we generate responses as $y_k^{(u)} = \left(\mathbf{x}_k^{(u)}\right)^T \boldsymbol{\vartheta}^{(j)} + \varepsilon_k^{(u)}$. The vectors $\boldsymbol{\vartheta}^{(j)}$ are sampled independently from a Gaussian distribution with mean $\mathbf{0}_p$ and covariance matrix $p^{-1/2}\mathbf{I}_p$.

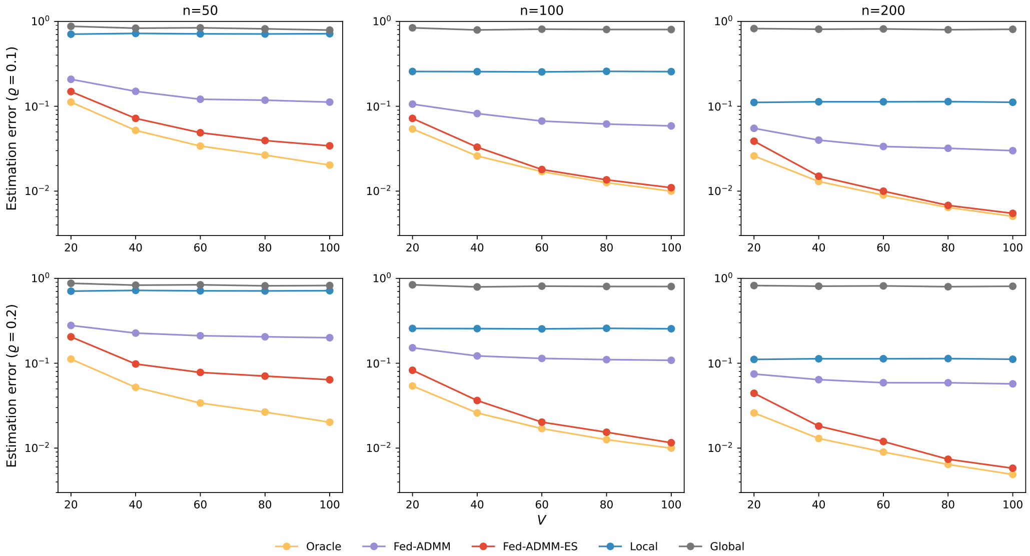

Figure 2: Average estimation error of FedADMM, FedADMM-ES and oracle, local and global estimators in linear regression, with randomly corrupted graphs. The number of clusters K is fixed to be 5 for all settings. The two rows correspond to corruption level $\varrho = 0.1$ and 0.2.

We generate $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by corrupting $\mathcal{G}_0$ at a specified *corruption level* $\varrho > 0$. This corruption involves randomly flipping the connection status between devices i and j in $\mathcal{G}_0$, guided by independent Bernoulli random variables e_{ij} with mean ϱ. In other words, if $\mathrm{e}_{ij} = 1$, we invert the connection status; otherwise, we leave it as it is. More specifically, we define the (i,j)th entry of the adjacency matrix by $\mathbf{\Lambda}_{ij}(\varrho) = \mathbf{\Lambda}_{ji}(\varrho) = \mathrm{e}_{ij}\{1 - (\mathbf{\Lambda}_0)_{ij}\} + (1 - \mathrm{e}ij)(\mathbf{\Lambda}_0)_{ij}$.

This applies to all $i, j \in \mathcal{V}$ with $i < j$. We then define $\mathcal{G}$ as the graph corresponding to $\mathbf{\Lambda}(\varrho)$. The degree of deviation between $\mathcal{G}$ and $\mathcal{G}_0$ can be modulated by different choices of ϱ. In fact, $|\mathcal{E} \setminus \mathcal{E}_0| + |\mathcal{E}_0 \setminus \mathcal{E}| = \sum_{i<j} \mathrm{e}_{ij} = \varrho V(V-1)/2 + O_P(\varrho V \log V)$, as confirmed by Hoeffding's inequality.

For estimation, we choose $m_u(\mathbf{z}_k^{(u)}; \boldsymbol{\theta}) = \big(y_k^{(u)} - \boldsymbol{\theta}^T \mathbf{x}_k^{(u)}\big)^2/2$ and $\phi(\cdot) = \|\cdot\|_1$ in (5). The regularization parameter λ is tuned by 5-fold cross-validation.

5.1 Estimation Error

We evaluate the average estimation error across various estimators, selecting V values from the set $20, 40, 60, 80, 100$, n values from $50, 100, 200$, with $K = 5$ and $p = 20$. We use corruption levels $\varrho = 0.1$ and $\varrho = 0.2$ for comparison.

Figure 2 illustrates the average squared estimation error for each estimator. The first and second rows correspond to corruption levels $\varrho = 0.1$ and $\varrho = 0.2$ respectively, with each data point representing the mean of 100 independent replications. In all settings, Local and Global performance does not vary with V, whereas the estimation error of Oracle decreases as V increases. With $\varrho = 0.1$, FedADMM outperforms local estimators, demonstrating the advantages of data federation. Nevertheless, a persistent performance gap between FedADMM and Oracle exists. Importantly, the average error of FedADMM does not diminish with an increase in V. As per Theorem 3, this occurs because as V grows, the expected number of incorrect edges increases due to the constant corruption level ϱ. The performance of FedADMM even falls below that of the local estimator with higher corruption, as shown in the second row of Figure 2.

Remarkably, edge selection effectively mitigates this issue. In all configurations, FedADMM-ES (FedADMM with edge selection) surpasses both FedADMM and the local estimator, with its performance nearing Oracle's when n is large. This implies that our edge selection process, detailed in Section 3.2, effectively eradicates most misleading information in the graph.

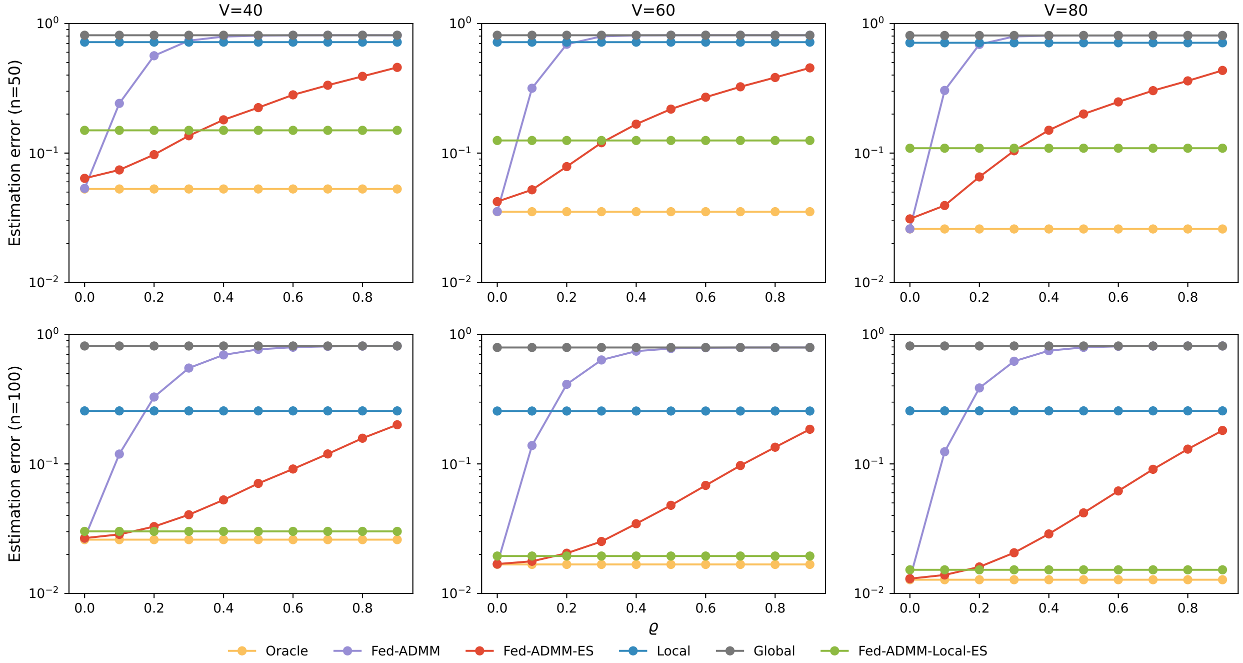

Figure 3: Average estimation error of FedADMM, FedADMM-ES, FedADMM-Local-ES, Oracle, Local and Global in linear regression, with randomly corrupted graphs. The x-axis corresponds to the corruption lever ϱ. We fix $K = 5$. The two rows correspond to $n = 50$ and 100.

5.2 Sensitivity Analysis of Graphs

We conduct an additional study to better understand the degradation in performance of FedADMM as ϱ increases. We vary ϱ from 0 to 0.9 in increments of 0.1 and compare the averaged estimation errors for FedADMM, FedADMM-ES, and FedADMM-Local-ES. We set $K = 5$, $V = 40, 60, 80$ and $n = 50, 100$. The results are displayed in Figure 3, with each data point reflecting the summary of 100 independent replications.

The average estimation error of the FedADMM increases rapidly with ϱ, exceeding that of the local estimator when $\varrho \geq 0.2$. In contrast, FedADMM-ES exhibits superior robustness, outperforming the local estimator even when the adjacency matrix is almost entirely misleading (i.e., $\varrho = 0.9$). Notably, FedADMM-ES performs better than FedADMM-Local-ES when ϱ is small, indicating that it benefits from accurate graph information, which is encapsulated by $\mathcal{G}$. However, the performance of FedADMM-Local-ES is not affected by ϱ, demonstrating its resistance to misleading graph information.

5.3 Algorithmic Convergence Rates

In this section, we investigate the algorithmic convergence rates of FedADMM and its variant, comparing them with both vanilla gradient descent (GD) and stochastic gradient descent (SGD). The latter two methods have convergence rates proportional to $1/\sqrt{T}$ in a convex but nonsmooth setting, where T is the number of iterations. For our experiments, we set $K = 5$, $p = 20$, and varied n over $50, 100$ and V over $40, 60, 80$. We ran FedADMM with a full batch, FedADMM with a batch size of 10, and both GD and SGD also with a batch size of 10. Figure 4 presents the results, with the x-axis representing the number of optimization iterations and the y-axis the average estimation error. In all configurations, FedADMM converged significantly faster than both GD and SGD.

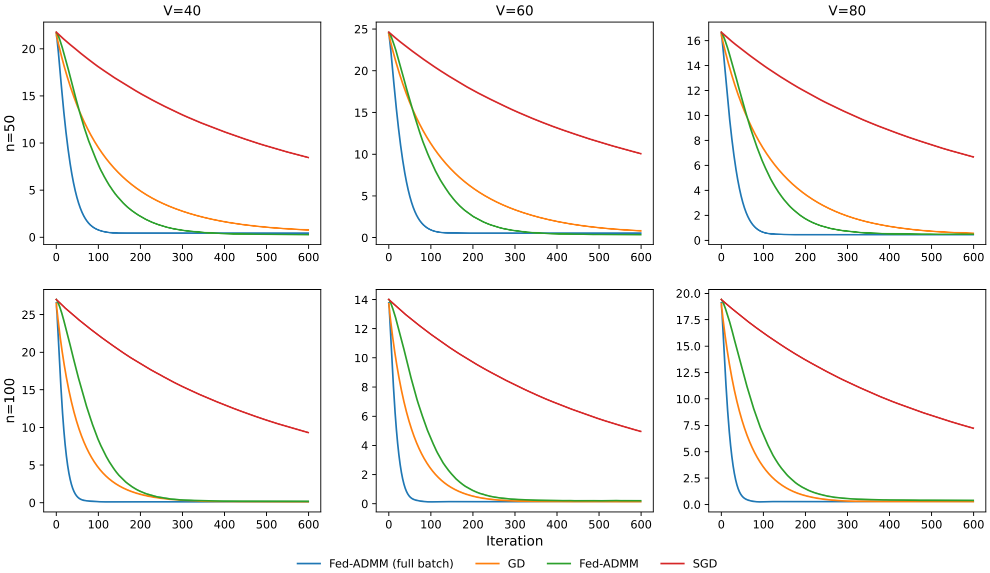

Figure 4: Learning curves of the FedADMM with full batch, FedADMM with batch size 10, GD and SGD with batch size 10. We set $K = 5$, $p = 20$, $n = 50, 100$ and $V = 40, 60, 80$. The x-axis and y-axis correspond to optimization iteration and average estimation error respectively.

6 A Real-Data Study

The significance of the 2020 US presidential election prompted us to apply our proposed method to examine its results. County-level election data were obtained from two publicly available sources, one providing election results that can be found in `https://github.com/tonmcg/US_County_Level_Election_Results_08-20` and the other offering county-level information that can be found in `https://www.kaggle.com/benhamner/2016-us-election`. This dataset encompassed 51 states, 3111 counties, and 52 county-level predictors.

We conceptualized each state as a device, with its counties acting as samples. The county-level information served as predictors, while the election result for each county was used as responses, encoded as 1 if Democrats won, and 0 otherwise. Logistic regression was then employed to predict election results. Owing to the scarcity of data, we included only states with more than 50 counties in our study, resulting in a total of 29 states.

The graph utilized in FedADMM was obtained through two distinct approaches: (a) using historical election results up to 2016 to classify states, whose incidence matrix is denoted by $\widehat{\mathbf{D}}_{\text{his}}$; and (b) using local estimators of states for edge selection, whose incidence matrix is labeled $\widehat{\mathbf{D}}_{\text{loc}}$. The first approach classified states based on their traditional political leaning as red (Republican), blue (Democratic), or swing states. The second approach relied solely on current local estimates to perform the edge selection procedure (11) with the given surrogate graph being a complete graph. Both local and global estimators were used as comparison benchmarks.

We divided the data, selecting 2/3 of counties at random for training, and using the remaining counties as a testing sample. The prediction accuracy was measured as the proportion of correctly classified samples over all test samples. This process was repeated 50 times, and the mean and standard deviation of each model's accuracies are presented in Table 1. As shown in Table 1, the best performance was achieved by FedADMM using $\widehat{\mathbf{D}}_{\text{loc}}$. The global estimator outperformed the local one, implying that heterogeneity among the states considered was not substantial. The performance of FedADMM with $\widehat{\mathbf{D}}_{\text{his}}$ was comparable

Table 1: Accuracy (mean(standard deviation)) of Local, Global, and FedADMM.

Methods	Local	Global	FedADMM	
			$\widehat{\mathbf{D}}_{\text{loc}}$	$\widehat{\mathbf{D}}_{\text{his}}$
Accuracy	0.741(0.034)	0.752(0.012)	0.793(0.019)	0.742(0.011)

to the local estimator, indicating that the heterogeneity did not principally stem from the traditional political leaning of states.

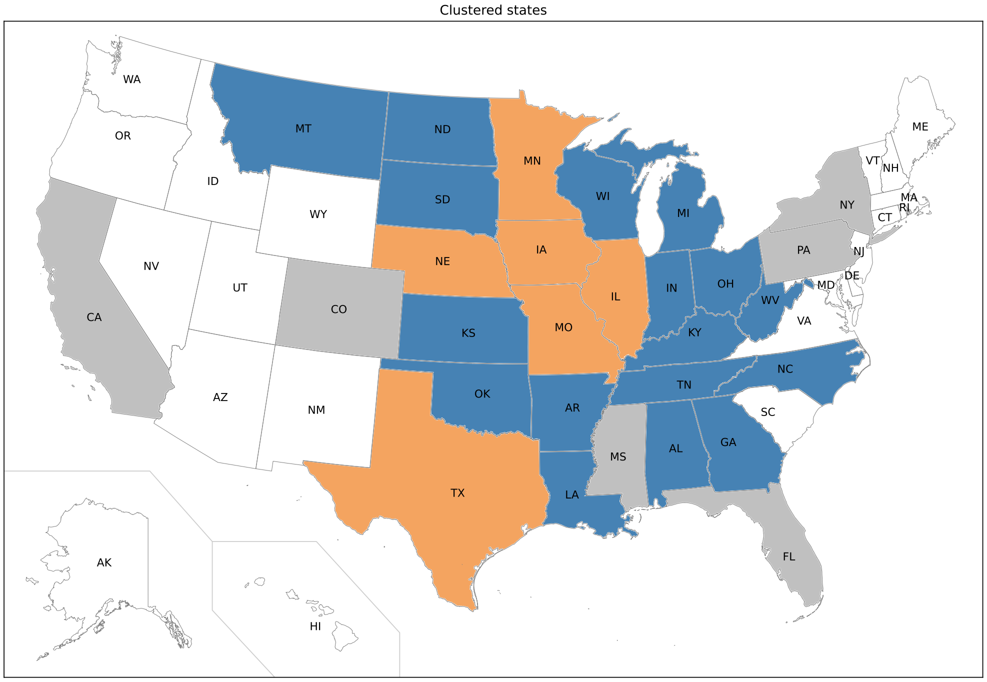

Figure 5: Clustering of states under FedADMM with $\widehat{\mathbf{D}}_{\text{loc}}$. Yellow and blue states represent two main clusters. Gray indicates states that are not connected to other states. White states are not considered.

In addition to prediction performance, we were also interested in the graph obtained by edge selection, which represents the similarity between states in the 2020 presidential election. The clustering result of the 29 states considered, derived using $\widehat{\mathbf{D}}_{\text{loc}}$ in our proposed method, is depicted in Fig 5. The graph consists of two connected components, i.e., two clusters. Members of these clusters are marked in yellow and blue, respectively, in Figure 5. This suggests that states within each cluster share similar electoral patterns. Some states, colored in gray, are not connected to any others in the graph. This could indicate that the statistical associations between predictors and electoral results in these gray states may diverge from the majority. As such, data from other states may offer little aid in predicting the results for these gray states.

7 Discussion

This study focuses on parameter estimation across multiple devices under significant constraints. These constraints include the prohibition of data sharing, heterogeneity of data distribution, limited computational capacity, and unstable accessibility of local devices. Within a broad M-estimation framework, we introduced a scalable and decentralized algorithm, and we established the convergence rate of our estimator under the Frobenius norm.

To achieve rate-optimality for our estimator, we require the surrogate graph to closely approximate the characteristic graph; that is, $K(\mathcal{G}) + |\mathcal{E} \setminus \mathcal{E}_0| = O(K(\mathcal{G}_0))$. While acquiring this prior information can prove challenging, we demonstrate that without any adjacency information of the characteristic graph, any estimator would perform no better than the local estimator. This phenomenon, known as the impossibility of adaptation, also arises in nonparametric multitask learning with covariate shifts (Hanneke and Kpotufe 2022). Our research expands on this by taking into account the heterogeneity resulting from varying target parameters. Despite these advancements, a comprehensive theory that simultaneously addresses shifts in covariates and drifts in target parameters remains an intriguing direction for future investigation.

While high-dimensional parameter estimation and inference are crucial, they have received scant attention in the federated learning literature. Recent research (Battey et al. 2018; Fan, Guo, and Wang 2021; Cai, Liu, and Xia 2021) has explored high-dimensional parameter estimation and inference in distributed settings without sharing local data. However, these approaches require the pre-computation of a local estimator from each device, which demands significant computational capacity. Our method has the potential to circumvent this issue by extending to high-dimensional settings under all the constraints typical of federated learning. In specific, we can achieve this by adding an additional ℓ_1-regularization of each parameter in (5), but the statistical inference in this case remains an open question for future research.

References

Accettura, N., Palattella, M. R., Boggia, G., Grieco, L. A., and Dohler, M. (2013), "Decentralized traffic aware scheduling for multi-hop low power lossy networks in the Internet of Things," in *IEEE 14th International Symposium on WoWMoM.*

Banerjee, M., Durot, C., and Sen, B. (2019), "Divide and conquer in nonstandard problems and the super-efficiency phenomenon," *The Annals of Statistics*, 47, 720 – 757.

Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. (2018), "Distributed testing and estimation Under sparse high dimensional models," *The Annals of Statistics*, 46, 1352–1382.

Benjamini, Y., and Hochberg, Y. (1995), "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 57, 289–300.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006), "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, 52, 2508–2530.

Bühlmann, P., and Van De Geer, S. (2011), *Statistics for high-dimensional data: methods, theory and applications*, Springer Science & Business Media.

Cai, T., Liu, M., and Xia, Y. (2021), "Individual Data Protected Integrative Regression Analysis of High-Dimensional Heterogeneous Data," *Journal of the American Statistical Association*, In Press.

Cui, Y., Che, W., Liu, T., Qin, B., and Yang, Z. (2021), "Pre-training With whole word masking for Chinese BERT," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 3504–3514.

Dobriban, E., and Sheng, Y. (2021), "Distributed linear regression by averaging," *The Annals of Statistics*, 49, 918–943.

Duan, R., Ning, Y., and Chen, Y. (2021), "Heterogeneity-aware and communication-efficient distributed statistical inference," *Biometrika*, 109, 67–83.

Fan, J., Guo, Y., and Wang, K. (2021), "Communication-efficient accurate statistical estimation," *Journal of the American Statistical Association*, In Press.

Fan, J., Wang, D., Wang, K., and Zhu, Z. (2019), "Distributed estimation of principal eigenspaces," *The Annals of Statistics*, 47, 3009–3031.

Fan, Z., and Guan, L. (2018), "Approximate ℓ_0-penalized estimation of piecewise-constant signals on graphs," *The Annals of Statistics*, 46, 3217 – 3245.

Gao, C., Han, F., and Zhang, C.-H. (2020), "On estimation of isotonic piecewise constant signals," *The Annals of Statistics*, 48, 629 – 654.

Godsil, C., and Royle, G. (2001), *Algebraic Graph Theory*, Springer New York.

Hallac, D., Leskovec, J., and Boyd, S. (2015), "Network Lasso: Clustering and optimization in large graphs," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Hankinson, R. J. (1987), "Causes and empiricism," *Phronesis*, 32, 329–348.

Hanneke, S., and Kpotufe, S. (2022), "A no-free-lunch theorem for multitask learning," *The Annals of Statistics*, 50, 3119–3143.

Hestenes, M. R. (1969), "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, 4, 303–320.

Hütter, J.-C., and Rigollet, P. (2016), "Optimal rates for total variation denoising," in *29th Annual Conference on Learning Theory.*

Jordan, M. I., Lee, J. D., and Yang, Y. (2019), "Communication-efficient distributed statistical inference," *Journal of the American Statistical Association*, 114, 668–681.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016), "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492.*

Krebs, M., and Shaheen, A. (2011), *Expander families and Cayley graphs: A beginner's guide*, Oxford University Press.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. (2020), "On the convergence of FedAvg on Non-IID data," in *Proceedings of the 8th International Conference on Learning Representations.*

Li, X., and Meng, X.-L. (2021), "A Multi-resolution Theory for Approximating Infinite-p-Zero-n: Transitional Inference, Individualized Predictions, and a World Without Bias-Variance Tradeoff," *Journal of the American Statistical Association*, 116, 353–367.

Mach, P., and Becvar, Z. (2017), "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys* & *Tutorials*, 19, 1628–1656.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. (2017), "Communication-efficient learning of deep networks From decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics.*

Ouyang, H., He, N., Tran, L., and Gray, A. (2013), "Stochastic alternating direction method of multipliers," in *Proceedings of the 30th International Conference on Machine Learning.*

Richards, D., Negahban, S., and Rebeschini, P. (2021), "Distributed Machine Learning With Sparse Heterogeneous Data," in *Proceedings of the 35th Annual Conference on Neural Information Processing Systems.*

Rockafellar, R. T. (1976), "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, 1, 97–116.

Sidransky, E., Nalls, M. A., Aasly, J. O., Aharon-Peretz, J., Annesi, G., Barbosa, E. R., Bar-Shira, A., Berg, D., Bras, J., Brice, A., et al. (2009), "Multicenter analysis of glucocerebrosidase mutations in Parkinson's disease," *New England Journal of Medicine*, 361, 1651–1661.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. (2017), "Federated multi-task learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems.*

Stich, S. U. (2019), “Local SGD converges fast and communicates little,” in *Proceedings of the 7th International Conference on Learning Representations.*

Strain, T., Wijndaele, K., Dempsey, P. C., Sharp, S. J., Pearce, M., Jeon, J., Lindsay, T., Wareham, N., and Brage, S. (2020), “Wearable-device-measured physical activity and future health risk,” *Nature Medicine*, 26, 1385–1391.

Suzuki, T. (2013), “Dual Averaging and Proximal Gradient Descent for Online Alternating Direction Multiplier Method,” in *Proceedings of the 30th International Conference on Machine Learning.*

Voigt, P., and von dem Bussche, A. (2017), *The EU general data protection regulation (GDPR)*, Springer International Publishing.

Wang, Y.-X., Sharpnack, J., Smola, A. J., and Tibshirani, R. J. (2016), “Trend Filtering on Graphs,” *Journal of Machine Learning Research*, 17, 1–41.

Wolfson, M., Wallace, S. E., Masca, N., Rowe, G., Sheehan, N. A., Ferretti, V., LaFlamme, P., Tobin, M. D., Macleod, J., Little, J., et al. (2010), “DataSHIELD: Resolving a conflict in contemporary bioscience—performing a pooled analysis of individual-level data Without sharing the data,” *International Journal of Epidemiology*, 39, 1372–1382.

Wooldridge, J. M. (2007), “Inverse probability weighted estimation for general missing data problems,” *Journal of Econometrics*, 141, 1281–1301.

Wu, J., Barahona, M., Tan, Y.-J., and Deng, H.-Z. (2011), “Spectral measure of structural robustness in complex networks,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41, 1244–1252.

Wu, Q., Chen, X., Zhou, Z., and Zhang, J. (2020), “FedHome: Cloud-edge based personalized federated learning for in-home health monitoring,” *IEEE Transactions on Mobile Computing*, 21, 2818–2832.

Zhang, X., Liu, J., and Zhu, Z. (2022), “Learning coefficient heterogeneity over networks: A distributed spanning-tree-based fused-Lasso regression,” *Journal of the American Statistical Association*, To Appear.

Zhang, Y., Duchi, J. C., and Wainwright, M. J. (2013), "Communication-Efficient Algorithms for Statistical Optimization," *Journal of Machine Learning Research*, 14, 3321–3363.

Zhao, T., Cheng, G., and Liu, H. (2016), "A partially linear framework for massive heterogeneous data," *The Annals of Statistics*, 44, 1400–1437.

Zhao, X., Wang, H., and Lin, W. (2023), "The Aggregation–Heterogeneity Trade-off in Federated Learning," in *Proceedings of Thirty Sixth Conference on Learning Theory.*